# Visual UI Testing

**10/10/2021 – 19/11/2021**

1. **~~Vision~~**
   1. ~~Basic process working~~
   2. ~~Advanced matching algorithms~~
      1. ~~dHash, pHash, aHash~~
      2. ~~SIFT, SURF~~
      3. ~~Siamese NN~~
   3. ~~Adaptive threshold~~
   4. ~~Modal recognition~~
   5. ~~Performance improvement~~
2. **~~Robotic~~**
   1. ~~Basic process working (Review and revise Robotic program) - 15/11/2021~~
   2. ~~Integration with Vision detection part~~

**22/11/2021 – 30/11/2021**

1. **~~Whole process improvement~~**
   1. ~~Accuracy~~
      1. ~~Vision~~
         1. ~~Detection~~
         2. ~~Matching~~
      2. ~~Robot arm~~
         1. ~~Calibration~~
   2. ~~Performance~~
      1. ~~Vision~~
         1. ~~Detection~~
            1. ~~OCR~~
         2. ~~Matching~~
      2. ~~Robot arm~~
2. **~~Robot Arm~~**
   1. ~~Hardware & driver upgrade~~
3. **~~Vision~~**
   1. ~~New matching algorithms~~
      1. ~~Auto-encoder~~
4. **Dataset**
   1. Auto-testing tools to click app & collect UI

**1/12/2021 – 12/12/2021**

1. **Dataset**
   1. GUI operation trace
      1. Set up Auto-testing tools to click app
      2. Collect UIs
   2. ~~Matching~~
      1. ~~Collect dataset~~
2. **~~Quantitative Evaluation~~**
   1. ~~Matching algorithms~~

**12/12/2021 – 19/12/2021**

1. **~~Matching Improvement~~**
   1. ~~Accuracy~~
      1. ~~Similar shape (area, aspect ratio)~~
      2. ~~Select the pair with max similarity~~
   2. ~~Speed~~
      1. ~~Pre-load model~~
      2. ~~Encode clips in batch~~
2. **~~Scenario Experiment~~**
   1. ~~Testing samples~~ 
      1. ~~First 2 steps~~

**19/12/2021 – 31/12/2021**

1. **Stabilize the system**
2. **Dataset**
   1. GUI operation trace (Benchmark)
      1. Set up Auto-testing tools to click app
      2. Collect UIs
3. **Empirical study design (scale & diversity)**
   1. Visual
   2. Testing strategy
   3. Hardware control
   4. Metrics
      1. Low level
      2. High level
      3. Correlation

**15/2/2022 – 25/2/2022**

1. **~~Visual approach~~**
   1. ~~Screen recognition~~
   2. ~~Element relative position based on screen~~
2. **Robot**
   1. Scroll operation
3. **Experiment for testing**
   1. Run auto testing tool
      1. Run the testing script
      2. Set collection strategies (threshold of number of operations)
      3. Collect dataset (small amount for testing)
   2. Robotic reply
      1. Same OS
      2. Cross platform
      3. Different device location
      4. Different robot arm
4. ***Paper draft***
   1. *Paper structure*

# Experiments

**Element Matching**

1. Detection-based
   1. SIFT
   2. HOG
   3. Resnet
2. Template matching-based
   1. CCOEFF
   2. CCORR
   3. SQDIFF
3. Textual content
4. Spatial information

**Record**

1. Sour APPs

**Replay**

1. Same OS
2. Cross platform
3. Different device location
4. Different robot arm

# Paper

**Critical Questions**

1. Who would benefit from the system?
2. What is the user scenario?
3. What is the overall architecture of the approach and system?
4. How to record?
5. Do we need specific language to define and create the test script?